

Raise The Bar: Video-Based Barbell Path and Velocity Tracking Application for Weightlifting

Neython Lec Streitz
nlecstreitz@oxy.edu
Occidental College

1 Problem Context

From the average gym-goer to the elite professional athlete, the ability to track and monitor training is key for supporting safe and effective workouts. Specifically for intermediate to advanced athletes, exercise efficiency and load management becomes crucial for maintaining long-term performance improvements. The purpose of this project is to provide an accurate, useful, and accessible method of tracking the path and velocity of a barbell during resistance training exercise via video-recorded exercises. Through the use of this application, athletes and coaches will be able to objectively track the range of motion and intensity of barbell-based exercises they perform.

Put simply, bar path refers to the range of motion of the barbell during an exercise. Bar path tracking thus, tracks the time and distance of the barbell during a specific lift. By optimizing the bar path for a given exercise, an athlete is able to pinpoint weak points in their movement and generate better efficiency overall. Especially for beginners, bar path is an incredibly helpful tool for improving and visualizing exercise form.

For the more advanced, velocity tracking provides even more granularity. Typically, exercise loads are prescribed using percentage-based methods. For example, a coach might make their athlete perform 5 sets of 5 repetitions using 70% of their 1-rep max. Percentage-based training has a long history of being an effective approach for resistance training design. Recently, Velocity Based Training (VBT) has emerged as a contemporary method of resistance training design, using the velocity of the barbell during an exercise to supplement and inform training decisions.

VBT is useful because it allows for more athlete-specific load (weight amount) and volume (repetition amount) prescriptions [27]. For one, it can gauge an athlete's current fatigue levels by comparing velocity in previous sessions. This practice is known as autoregulation, where an athlete can pull back on high fatigue days and push harder on days they are feeling fresh. Secondly, VBT allows coaches (or self-lead gym-goers) to assign training loads that deliver more precise and objective levels of effort and fatigue. Some athletes have difficulty doing higher repetition work,

whereas others see sharp drop-offs near their one repetition max. In the percentage-based approach, set and repetition recommendations do not account for this variability [1]. With all that in mind, bar path and velocity metrics serve as a way to standardize technique for a beginner, and for the advanced, give biofeedback in real-time and enable the design of athlete-specific programs.

Currently, there exists various hardware and software-based approaches to tracking bar path and velocity. This includes hardware devices like linear position transducers and 3D motion analysis systems, as well as video-based smartphone apps. However, using hardware to track velocity is expensive and inaccessible for the majority of regular gym-goers and even for advanced athletes. Furthermore, most smartphone apps show poor accuracy and clunky design [18, 13]. Thus, our project is in a position to improve on the accuracy and design of existing smartphone apps, while circumventing the money and infrastructure needed for hardware-based approaches.

2 Technical Background

At its most basic, the question at hand is how we can track, and ultimately display, the position and rate of change of a dynamic object in real time. Thus, this project is one in the realm of computer vision (CV), or, how computers interpret the visual world. At a glance, the technical components of this project include object detection and object motion tracking.

2.1 Object Detection

To begin, there needs to be a way to detect the barbell within the frame of the video. The main goal of object detection within CV tasks is to identify and draw boundaries around objects in a given image or video. State-of-the-art approaches utilize deep-learning methods and neural networks to perform object detection. To this end, there are typically two main parts to object detection with deep-learning. Firstly, the image or frame passes through an encoder that extracts statistical features used to locate and label objects. Then, the encoder's outputs are passed through

a decoder that predicts the boundaries and labels of each of the objects.

An existing network architecture is usually the encoder used for object detection. Two of the most common network architectures are VGG and ResNet [24, 9]. However, these networks are simply too large for resource constrained devices like smartphones. Developed by Google, MobileNet is an encoder specifically made to work on mobile devices [11]. It functions by employing depthwise separable convolution, a process that decreases the amount of parameters in the network, making it less resource intensive.

Common decoders used for object detection applications include Faster R-CNNs and You Only Look Once (YOLO) [22, 21]. While Faster R-CNNs delivers great accuracy, it suffers in overwhelming complexity and slow speed. YOLO has the opposite problem, as it provides speed but lacks in accuracy. Google's Single Shot Detectors (SSD) aims to provide a middle ground [17]. SSD works by laying a grid of anchor points over the input image then overlaying boxes of multiple shapes and sizes at each anchor point. The model then predicts the likelihood that the object to be detected exists within the box, changing the boxes to fit the object as best as possible. The lower likelihood boxes are pruned away, leaving a best prediction of the boundaries of the object.

By combining MobileNet and SSD, smartphones are capable of supporting deep learning-based object detection. Having said that, the MobileNet and SSD combination supports the detection of 20 different objects, none of which are barbells. Given that the process of training a network to specifically detect barbells is likely very GPU intensive, slow, and unreliable, my project takes advantage of the relative ease of tracking a pre-specified color and shape. Specifically, I will take advantage of pre-existing fiducial marker detection techniques. Fiducial markers, or in other words, QR codes, are incredibly great for estimating camera pose. The corners of a QR code provide the necessary correspondences to estimate the position and orientation of the barbell. Furthermore, because of their inner codification, the QR codes add extra reliability in detection. Combined with the fact that barbells are standardized sizes, adding fiducial markers to object detect, essentially turns the barbell into a contour.

A contour is a well-defined boundary with typically matching color or intensity within. In the case of fiducial markers, there is a bit pattern within the outline. To allow for contour detection, the image or video frame is first converted to grayscale. Next, binary thresholding is applied, converting the image to black and white by setting only the borders of the object white and the rest black. A contour algorithm is finally applied to identify the borders, and if needed, draw them out over the original image. The algorithm simply detects potential shapes that match the outline

of our contours. Once detected, the contours are analyzed by extracting the marker bits within the marker. A perspective transform is applied to isolate the marker. The image is further divided into separate cells and bits within each cell are counted. The bit counts are matched against a dictionary to determine validity as a marker. Contour detection is thus a much simpler process, allowing me to maintain accessibility and ease-of-use, without the need to train a novel neural network.

2.2 Object Tracking

In essence, the velocity of a barbell is simply its directional movement over time. To get the barbell velocity, it is necessary to track it throughout the frames of a video. In the case of a barbell, we will be engaging in Single Object Tracking (SOT).

Typical problems in object tracking include occlusion, detection speed, and spatial scale variation. Occlusion is simply when objects are so close to each other that they appear to be one merged object. Luckily, supporting occlusion is not something we need to deal with, as the barbell will almost always be in an open and unimpeded space. Similarly, because we are tracking the barbell side-on, almost as if it was two-dimensional, there is little variation in object scale. Detection speed is certainly an issue, but because this is a contour detection problem, there already exists fast and efficient techniques.

There are a variety of neural network approaches to object tracking, including BOOSTING Tracker, MIL Tracker, and KCF Tracker [8, 4, 10]. These approaches each have their benefits and downsides, typically balancing general accuracy, occlusion support, lighting and viewpoint change, and error detection.

For the purposes of this project, tracking will involve repetitively updating the camera and calculating the previous pixel position with the current. Knowing the size of the end of the barbell, the amount of pixels occupied by the barbell in the frame can be used to track the movement of the barbell throughout the video. A list-like data structure such as the *deque*, provides quick enough pops and appends to maintain and retrieve the previous positions of the barbell. Due to the frame rate of modern smartphones, this process should be quick enough to accurately track the position and velocity of the barbell.

3 Prior Work

As mentioned before, there currently exists a variety of hardware and software-based methods of tracking barbell path and velocity. While the market is fairly large, there are only a handful of devices, systems, and apps that have been studied in sports science literature. The studies that have

been done, mainly evaluate the reliability, ease-of-use, and practicality of these technologies.

Before discussing explicit barbell path and velocity tracking projects, it is important to mention works related simply to fiducial marker detection. Prime among these include work out of the University of Michigan by Wang et al., where they developed an improved algorithm for AprilTag detection. AprilTags are specialized fiducial markers used in many robotics applications because of their utility in camera pose estimation. They are black and white square tags with an encoded binary pattern. The improved algorithm takes advantage of the rarity of occluded tag detection, allowing the algorithm to function quicker. Furthermore, Wang et al., showed detection improvements at a wide variety of distances away from the tags. The AprilTag system is based on even earlier work including ARTag and ARToolkit [14, 7]. Ultimately, these fiducial tag systems and detection algorithms are efficient, applicable in a wide variety of CV problems, and maintain high accuracy and resilience against false positives.

As far as hardware-based approaches are concerned, linear position transducers, also known as linear encoders, are considered the gold standard for barbell velocity tracking. These devices clip onto the barbell and have a cable moving in and out of the device. Examples of linear encoders available on the market include Vitruve, T-Force, and GymAware. These companies also include or offer paid mobile apps for synchronization with their devices. All three of the listed devices have been validated by studies in sports science literature [18, 25, 19]. However, great accuracy comes at a price, as linear encoders are expensive and inaccessible. The lone GymAware device comes in at over \$1,995, with accompanying software starting at an additional \$495 minimum. Vitruve is a cheaper option, yet still comes in at around \$397 for the device and mobile app. One more thing to note is the potential for damaging the devices because they must remain directly under the barbell. This is especially problematic for intensely dynamic movements like the Olympic lifts.

There are currently no hardware-based devices on the market with the express purpose of visualizing barbell paths. That's not to say that three-dimensional motion capture camera systems cannot and have not been used to track bar path and velocity [19]. Yet, like linear encoders, camera systems are costly and involve infrastructure not accessible nor feasible to set up for the average gym-goer. Nonetheless, the reliability of these devices means they are often used to test the validity of novel velocity trackers, like mobile apps [19].

For software-based approaches, specifically, mobile applications, there are multiple offerings on the market. Examples of apps that include both bar path and velocity tracking include MyLift, IronPath, and BarSense. Of those,

MyLift and IronPath have been studied in the literature. Markedly, studies have concluded that both MyLift and IronPath are not reliable tools for the measurement of barbell velocity. Kasovic et al. showed that the IronPath app compared with a linear transducer, "recorded significantly lower average concentric velocity values for the front squat and back squat and greater ROM values for the sumo deadlift" [13]. Similarly, Martinez-Cava et al. concluded that "the excessive errors of the newly updated MyLift app advise against use of this tool for velocity-based resistance training" [18]. With that said, other apps as well as experimental software approaches have shown promising results.

In a paper by Pueo et al., they developed a smartphone-based camera system for automatically detecting and tracking a barbell using a custom image-processing algorithm [20]. The algorithm works by creating reference points on the parallel supports of the exercise machine, in this case, a Smith Machine, to segment and compute the barbell position without prior calibration. Their system showed near matching performance with a linear position transducer. That being said, their paper also admitted that the video system is limited by the use of a barbell machine, and does not support free weight barbell movements. Moreover, their system does not include any path tracking and visualization features. Nonetheless, in terms of velocity tracking accuracy, their paper is certainly something to look up to. For the purposes of this project, the main takeaway from this study is the use of colored tape on the barbell to improve accuracy.

Interestingly, two papers by Balsalobre-Fernández et al. concluded that the MyLift (formerly Powerlift) app was in fact a valid and reliable method of tracking barbell velocity [2, 3]. The two studies showed high correlation between MyLift and a linear position transducer. However, a study and subsequent response by Courel-Ibañez et al. contradicted these findings, showing poor reliability from the MyLift app [5]. They also noted that the main author of the two papers where MyLift was validated, is also the main developer of the app. Combined with the aforementioned study by Martinez-Cava et al., that also advised against its use, the MyLift app does not serve as an example of a validated smartphone-based velocity tracker [18].

Ultimately, while linear position transducers deliver incredible accuracy and precision, they are far too expensive to be accessible tools for regular gym-goers and most athletes to use in the gym. Not to mention that linear position transducers also lack in the visualization of barbell path and have the danger of being damaged by the barbell. Unfortunately, due to their consistency, very little work is being done to improve on their flaws. Otherwise, several mobile apps already exist on the market that offer both barbell path and barbell velocity tracking for free. Yet, their inability to track velocity with accuracy and precision makes them

unreliable for the practice of velocity based training methods. Pushing this field of technology forward means developing apps with sufficient precision to implement VBT, while maintaining ease-of-use, quality design, and accessibility to a wide audience of gym enthusiasts and athletes.

4 Ethical Considerations

For this project, there exists various ethical considerations to be weary of, especially with the resources currently available for this project. These issues are directly linked to the main goals of the project. That is, providing barbell tracking accurately, increasing accessibility by lowering associated costs, and allowing tracking to occur over time. Primary among these considerations are the potential of providing faulty tracking to users and privacy issues concerning maintaining user video data for future review. Individually, these potential issues can be addressed, but as a whole, they might entail spending too much time on disparate features. Moreso, the current development plan of the project involves building solely the velocity and path tracking model before implementing it into a user application. This means that ethical concerns with app design will not be a main concern in this beginning stage, which could cause problems in the future. As such, we spell out the two primary concerns below.

4.1 Providing Accurate Velocity

Paramount to the utility of the project is the need for accurate tracking of bar path and bar velocity. Inherent with VBT methods is the fact that slight variances in velocity indicate important information about neuromuscular and functional performance [6]. It is these changes that velocity tracking hopes to analyze. Thus, one of the largest potential issues with the project is that large enough deviations from true velocity scores have the potential to invalidate the training method.

While deviations on the lower end might only lead to under-prescribing lifting intensity, deviations on the higher end have the potential to injure lifters relying on the application. When speaking on the ethical considerations of AI-based health-related apps, researcher Michael Kühler argues that "AI health apps... arguably lead to the novel issue of AI paternalism, particularly in the health care domain". That is, health and fitness apps are an instance of "persuasive technology" whereby the app relies on strategies (gamification, positive feedback, etc.) to motivate users towards specific ends [16]. The largest criticism of paternalism is the idea that it relies on an external notion of what is good for a person. A main advantage of VBT over traditional percentage-based training is the added personalizability that comes with basing training on daily and historic metrics.

Nonetheless, this added personalizability gives VBT the allure of objectivity when it comes to prescribing exercise volume and load, meaning dedicated lifters or even novices might force themselves to perform more than they can handle. This can lead to problems of injury, overtraining, or burn-out, all issues that are contrary to the stated goal of the app.

The fact of the matter is that these issues do not exist if the app can provide sufficient accuracy as well as present the information in a way that maintains a user's autonomy. However, the idea of providing necessary accuracy is much easier said than done. Currently, there exists no mobile applications on the market that have been fully validated by existing scientific literature: "While there is conflicting evidence, it appears that substantial bias and error can be introduced when different devices and/or users implement these measuring tools" [27]. Therefore, it is unlikely that our project will provide accurate enough velocity metrics to be reliably used for comprehensive velocity-based programming. In total, this means we must be careful about the framing of the project and its velocity output and recommendations.

4.2 Privacy of Video Data

Certainly, seeing bar path and velocity metrics on a given day allows lifters and coaches to modify and improve their daily resistance training programming. Even more useful is tracking those metrics across time, as a lifter/coach gains an understanding of a lifter's progression, performance compared to previous sessions, and patterns in training. So, our project aims to deliver a feature based on providing and comparing metrics across time. However, this comes with its own issues and it now requires the application to store user data. This might include videos of exercises being performed, velocity metrics per day, or simplified bar path graphs. As a result, there are new ethical considerations in terms of privacy concerns.

While exercise video data is not a typical form of health data, it still must be kept privately. As Kühler points out "even if AI health apps' paternalistic influence on users were beneficial for them and did not undermine or diminish their autonomy, such apps might still end up being considered ethically problematic" [16]. Barbell tracking would stand to improve by tapping into user data. As our application would store the videos and metrics of our users, there is a potential for a substantial repository of data to pull from. Yet, this is exactly the issue with storing data, specifically, health and fitness data. As an application, users need not just be people interested in weight lifting. Perhaps, a user might be a physical therapist using the application to help their patients improve their movement. This would truly mean the application is storing health data. As a result, us-

ing user data will require informed consent and opacity on our end as developers. Additionally, using our application in a gym setting might inadvertently cause non-participating people to be recorded within the video. This is especially problematic because those recordings would be without the person’s knowledge and there would be virtually no way to gain that informed consent. Some gyms even forbid video recording outright without prior permission from the staff. As such, our application sits in a difficult spot between providing high-granularity features and maintaining user and public privacy.

Less outwardly questionable is the issue of simply keeping the data private from external sources. To mitigate this potential problem, Schaar recommends employing, *Privacy by Design* including minimizing data collection, data sovereignty, and right to information among a handful of other concerns [23]. There does not seem to appear any potential features of the application that would deny these recommendations, but it will lead to longer development time. Questions we should be asking including: Does our application need a password feature? Should we store the entire video or just a reduced bar path representation? Should users be able to export and share their videos with others? These questions are somewhat difficult to address considering the projected development of the application starts not with front-end design but by focusing on creating highly accurate bar path and velocity tracking features. Regardless, storing health data of app users will necessitate careful consideration in terms of data usage and data privacy.

5 Methods

As a problem of computer vision, the majority of the technical aspects of the project have to do with identifying and then tracking the barbell. As mentioned before, we have chosen to use QR code-like barbell tags which are easily printable and placed on a wide variety of barbells. This is cheap, removes the need to color and/or permanently mark a barbell (that might be for public use), and has been used for other CV applications with great success. Velocity and bar path tracking involves the constant identification and optical flow estimation of the bar over time.

5.1 Barbell Identification

Based on a previous project by the April Robotics Laboratory out of the University of Michigan, our barbell identification will rely on QR code (fiducial marker) identification [26]. Why QR codes? As low cost is a primary goal of the project, by using QR codes we can remove the need for training novel machine-learning-based barbell identification models while still maintaining increased accuracy. QR codes are easy to generate, cheap and printable, and

more efficient to stick/remove from barbells than permanent options (i.e, coloring the barbell). We also considered using brightly colored stickers as our contour, but the use of QR codes nearly guarantees that the correct contour is identified within the frame because of their inner codification. Thus, they are a great candidate for barbell identification.

With the QR codes in mind, the first step is identifying the barbell. We will start by printing a QR code based on work by Krogius et al., where they propose a flexible model for generating fiducial markers: “We can generate layouts with a higher data density and smaller border, circular tags, or a custom tag layout with empty space in the middle in which a smaller tag could be placed, allowing this sort of recursive tag to be detected over a large range of distances” [15]. This means we can generate QR codes that fit nicely onto the circular end of a barbell, which typically follow standard dimensions, or at the very least offer easily accessible dimensions in their manuals. For a user, the project will generate and email them a QR code image for them to print. Currently, a user will be able to select a QR code to fit the shaft dimensions of a 2.8cm standard men’s Olympic barbell or a 2.5cm standard women’s Olympic barbell. The QR code will be taped centrally onto the end of the barbell facing the camera.

After starting the video recording, the QR code detection follows with two main steps. Firstly, the image will be processed via adaptive thresholding, segmenting the current frame by pixel intensity. Contours will then be extracted from this thresholded image, with special attention paid to only those contours that match the general outline of our marker. This processing essentially encompasses marker candidate detection. Secondly, to determine whether or not the candidates are truly markers, their inner codification will be verified. The bits of each marker will be extracted, counted (white and black bits), and then checked against a dictionary to determine validity. Fortunately, the barbell will be recorded directly parallel, meaning no extra work needs to be done to determine orientation or solve for occlusion.

5.2 Bar Velocity and Path Tracking

By repeatedly identifying and locating the QR code frame-by-frame, we should be able to track the barbell over time, giving us a path and velocity output. The algorithm will begin by initializing, waiting for a QR code to appear in the scene. Since the camera will already be set up to view the barbell and QR code, this process will happen rather quickly. We will take that first frame as our original reference. Next, we will perform QR code detection as outlined in the above subsection, saving the QR code bounding rect. Continuing, we will perform feature matching, that is, we will identify the differences in the original frame to next

frames, and estimate the optical flow between the frames. Optical flow is used once again to estimate the bounding rect position of the QR code in the next frame. Given that the diameter of our barbell's end is 50mm and each frame is 1/fps seconds in length, the pixel distance frame to frame along with the amount of pixels occupied by the QR code (barbell's end) should be enough to calculate mm/s every 1/fps. Lastly, we will repeat this process continually until the user ends the video.

The contrail of the barbell (the bar path) is relatively straightforward to implement. During the bar localization process, we will save the past and current points and as long as they are not null, we will compute the thickness and draw a line connecting the points. Velocity tracking similarly takes the past and current location points to determine distance over time. Bar position and velocity metrics will be saved continually to be outputted intra-set and for viewing after a set.

6 Evaluation Metrics

The stated goals of the project are to deliver a cheap, accessible, and accurate velocity and path tracking application for barbell-based exercise. On the whole, QR codes are quite inexpensive to print and only require some form of adhesive to place on a barbell. Thus, the resources necessary to use the velocity tracking are far less expensive and far more accessible than a linear position transducer. However, our project is slightly more expensive and slightly less accessible than smartphone apps on the market, given that a QR must be printed, whereas those apps require nothing but a phone. Nonetheless, this added cost should come with a much needed increase in accuracy compared to the smartphone apps on the market. We delineate the exact accuracy evaluation metrics in the velocity accuracy subsection. Additionally, we will evaluate the ease-of-use and utility of the app via one round of user testing. The specifics are discussed in the app utility subsection.

6.1 Participant Selection

We will select fifteen users to participate in a joint evaluation on app utility and app accuracy. Users will be made aware of the training design and procedure. Users will also be informed of the basic advantages of using velocity-based training methods. Our hope is that no participant will report any injuries or limitations prior to beginning the study. All participants will be familiarized with the exercises used in the study. Lastly, users will sign a written informed consent form.

6.2 Study Design

Participants will utilize the app for two different exercises: barbell back squat and barbell bench press. Prior to starting, participants will fill out a pre-study questionnaire. After finishing the survey, participants will begin with the active portion. For each of the two exercises, participants will perform a progressive loading test consisting of increased loads of 15 lbs starting and ending in a range of between 45 lbs and 135 lbs. After each set, participants will take 5 minutes of rest to recover and analyze their bar path and velocity metrics outputted from the app. The magnitude of errors and levels of agreement between the app and a gold standard LPT will be calculated for each set for each user.

6.3 Study Methodology

After a familiarization session with the exercises and barbell, participants will be allowed time to browse through the app. Participants will be asked previously to give their one-repetition maximum to ensure all participants can complete the study with each given weight. The following pre-study survey will be expected to be completed by each participant:

1. How familiar were you with velocity-based training prior to the study?
2. Have you used a different velocity/bar path tracking app in the past?
3. How often do you train barbell-based exercises on a weekly basis?
4. What issues do you have with your current training?
5. What features would you expect from a velocity-based training app?

Next, participants will be asked to perform three repetitions at maximum velocity for each weighted increase with 5 minutes of rest in between. Participants will be allowed to set up the video recording as they like following hands-off guidance mentioned by the application. The video recording will be done using a basic iPhone smartphone. In between sets, participants will be allowed to pick up the smartphone to simulate real-world recording and test the project's adaptability in different settings. During this time, we will record the output from the LPT and after the participant is done, from the smartphone as well (to be uploaded to the app). After completing the study, the following post-study questionnaire will be presented to the participants:

1. If you've used a similar app before, how would you compare your experiences?

2. What parts of the app did you find useful?
3. What parts of the app did you NOT find useful?
4. What parts of the app did you find difficult to use?
5. Would you use the app in the future? Why or why not?
6. What is your perception of velocity-based training after this study?
7. Are there any additional comments?

6.4 Velocity Accuracy

To evaluate the accuracy of our app, we will follow standard convention and employ the use of a linear position transducer. As stated by Janicijevic et al., LPT's, specifically GymAware and T-Force have been validated multiple times in the literature "as gold-standards in studies designed to validate other velocity monitoring devices"[12]. As such, we will compare our app to a gold-standard device.

Reliability analysis will be completed by calculating the levels of agreement and magnitude of error between the LPT and app for each participant. To do so, we plan on creating Bland-Altman plots. These plots will be scatterplots of the difference between the velocity output of the app and of the LPT compared to the average of the two measurements. The magnitude of error will be calculated by taking the difference between the app's output and the LPT and dividing that difference by the LPT velocity measure.

The reason for the gradual increase in barbell weight is to ensure reliability is covered for a variety of velocity ranges. Similar, but more precise, work was done by Martinez-Cava et al. [18]. Correlation will be calculated using the Pearson's correlation coefficient and the intraclass correlation coefficient (ICC). Based on the aforementioned work by Martinez-Cava et al., we will set our acceptable reliability at $ICC > 0.990$ [18].

6.5 Project Utility

Project utility will be analyzed using the pre and post-survey results after study completion. Surveys will be kept anonymous and after reading responses, key themes and re-occurring mentions will be recorded from the responses. Importantly, our study will miss a few key determinants of project utility. A longitudinal study might reveal issues of adherence, day-to-day bugs, or potential future features. Unfortunately, a study over a larger period of time is out of scope for this project. As a result, we hope that this study will serve as a gauge of initial utility and interest in VBT, as well as identify any glaring problems with user-app interaction.

7 Timeline

Below, I have laid out a general timeline for each component of the project. The idea here is to focus on more complex features before incorporating any user interface or quality-of-life features. With that said, the timeline is as follows:

- **Summer (May - Early August)**

Conduct further research to investigate potential features and obstacles for project development. These include further research into why currently existing applications cannot sufficiently accurately track velocity and what proponents of VBT need/want in a product.

- **Semester Start (August 25th-September 6th)**

Develop QR code generation feature. Even though this will be easier than the following few features, it is a necessary first step.

- **Early Semester (September 7th-September 30th)**

Develop velocity tracking feature. This is easily the most complex and most instrumental feature of the entire project. As such, I plan on spending a large amount of time developing the velocity tracking. If necessary, this step can leak into October. It involves QR code detection and tracking.

- **Middle Semester (October 1st-October 20th)**

Develop barbell path tracking feature. This is a complementary feature to velocity tracking and as such, it goes after the velocity tracking is complete. I initially believe this feature will be much less complex than velocity tracking, especially because it requires much less accuracy.

- **Late Middle Semester (October 21th-November 15th)**

Prepare poster to be turned in on November 15th. This time frame will also be used to conduct project evaluations. Depending on the state of the project, this might just be accuracy testing or user testing. If I am in a good place by this point, I can consider adding extra quality-of-life/user interface features.

- **Late Semester (November 16th-December 5th)**

Continuing implementing previously evaluated issues, missing/superfluous features and prepare for presentation. During this time I will fix any potential issues with the project, evaluate missing or superfluous features, and potentially plan for a following semester of work. I should be completely prepared to showcase and present on my final project by December 5th if not

a few days before. This time will also be used to prepare my final paper.

- **End of Semester (December 5th-15th)**

Finish final paper and prepare entire project to be submitted. This includes any missing code features, cleaning up code, and making software documentation clear and robust. The goal is to be evaluating minor syntax, clarity, and wording issues by this point and not be focusing on any major changes. All resources will be submitted by December 15th.

The exact final product is still not entirely set. What I mean by that is that whether or not my final project includes user interface features depends on how quickly I am able to develop the velocity and bar path tracking features. Without robustness in these features, the user interface components are essentially meaningless. Thus, the timeline as stated now assumes the "worst case" in which only the major components are developed. The plan now is to publish the project as a smartphone application during a following semester. All in all though, I feel that my project is quite concrete and I am confident in my ability to develop and present it by the December 15th deadline.

References

- [1] Balsalobre-Fernández, C and Torres-Ronda, L. "The Implementation of Velocity-Based Training Paradigm for Team Sports: Framework, Technologies, Practical Recommendations and Challenges". In: *Sports (Basel)* (2021).
- [2] Balsalobre-Fernández, C et al. "Analysis of Wearable and Smartphone-Based Technologies for the Measurement of Barbell Velocity in Different Resistance Training Exercises". In: *Frontiers in Physiology* (2017).
- [3] Balsalobre-Fernández, C et al. "Validity and reliability of a novel iPhone app for the measurement of barbell velocity and 1RM on the bench-press exercise". In: *Journal of Sports Sciences* (2018).
- [4] Boris, B, Yang, M, and Belongie, S. "Visual Tracking with Online Multiple Instance Learning". In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2009).
- [5] Courel-Ibáñez, J. et al. "Technical Note on the Reliability of the PowerLift App for Velocity-Based Resistance Training Purposes: Response". In: *Annals of Biomedical Engineering* (2020).
- [6] Dorell, H.F, Smith, M.F, and Gee, T.I. "Comparison of Velocity-Based and Traditional Percentage-Based Loading Methods on Maximal Strength and Power Adaptations". In: *J Strength Cond Res.* (2020). URL: <https://pubmed.ncbi.nlm.nih.gov/30946276/>.
- [7] Fiala, M. "ARTag, a fiducial marker system using digital techniques". In: *Computer Vision and Pattern Recognition* (2005).
- [8] Grabner, H, Grabner, M, and Bischof, H. "Real-Time Tracking via On-line Boosting". In: *Proceedings of British Machine Vision Conference (BMVC)* (2006).
- [9] He, K et al. "Deep Residual Learning for Image Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).
- [10] Henriques, J et al. "High-Speed Tracking with Kernelized Correlation Filters". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2015).
- [11] Howard, A et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". In: *ArXiv* (2017).
- [12] Janicijevic, Danica et al. "Comparison of the two most commonly used gold-standard velocity monitoring devices (GymAware and T-Force) to assess lifting velocity during the free-weight barbell back squat exercise". In: *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology* 0.0 (2021), p. 17543371211029614. DOI: 10.1177/17543371211029614. URL: <https://doi.org/10.1177/17543371211029614>.
- [13] Kasovic, J et al. "Agreement Between the Iron Path App and a Linear Position Transducer for Measuring Average Concentric Velocity and Range of Motion of Barbell Exercises". In: *Journal of Strength and Conditioning Research* (2021).
- [14] Kato, H. and Billinghurst, M. "Marker tracking and hmd calibration for a video-based augmented reality conferencing system". In: *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on. IEEE* (1999).
- [15] Krogius, M., Haggemiller, A., and Olson, E. *Flexible Layouts for Fiducial Tags*. 2019.
- [16] Kühler, M. "Exploring the phenomenon and ethical issues of AI paternalism in health apps". In: *Bioethics* (2022). URL: <https://doi.org/10.1111/bioe.12886>.
- [17] Liu, W et al. "SSD: Single Shot MultiBox Detector". In: *Computing Research Repository (CoRR)* (2015).

- [18] Martínez-Cava, A et al. “Reliability of technologies to measure the barbell velocity: Implications for monitoring resistance training”. In: *PLOS One* (2020).
- [19] Pérez Castilla, A et al. “Reliability and Concurrent Validity of Seven Commercially Available Devices for the Assessment of Movement Velocity at Different Intensities During the Bench Press”. In: *The Journal of Strength and Conditioning Research* (2019).
- [20] Pueo, B et al. “Video-Based System for Automatic Measurement of Barbell Velocity in Back Squat”. In: *Sensors (Basel)* (2021).
- [21] Redmon, J et al. “You Only Look Once: Unified, Real-Time Object Detection”. In: *CoRR* (2015).
- [22] Ren, S et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *CoRR* abs/1506.01497 (2015).
- [23] Schaar, P. “Privacy by Design”. In: (2010). URL: <https://doi.org/10.1007/s12394-010-0055-x>.
- [24] Simonyan, K and Zisserman, A. “Very Deep Convolutional Networks For Large-Scale Image Recognition”. In: (2015).
- [25] Wadhi, T et al. “Validity and Reliability of the GymAware Linear Position Transducer for Squat Jump and Counter-Movement Jump Height”. In: *Sports (Basel)* (2018).
- [26] Wang, J. and Olson, E. *AprilTag 2: Efficient and robust fiducial detection*. 2016.
- [27] Weakley, J et al. “Velocity-Based Training: From Theory to Application”. In: *Strength and Conditioning Journal* (2021).